

Secretive Coded Caching

Vaishakh Ravindrakumar*, Parthasarathi Panda[†], Nikhil Karamchandani[‡], and Vinod Prabhakaran[§]

* Dept. of Electrical and Computer Engg., UC San Diego, Email: vaishakhr@ucsd.edu

[†] Amazon India, Email: parthasarathipanda314@gmail.com

[‡] Dept. of Electrical Engg., IIT Bombay, Email: nikhilk@ee.iitb.ac.in

[§] School of Technology and Computer Science, TIFR, Email: vinodmp@tifr.res.in

Abstract

Recent work by Maddah-Ali and Niesen (2014) introduced *coded caching* which demonstrated the benefits of joint design of storage and transmission policies in content delivery networks. They studied a setup where a server communicates with a set of users, each equipped with a local cache, over a shared error-free link and proposed an order-optimal caching and delivery scheme. In this paper, we introduce the problem of *secretive coded caching* where we impose the additional constraint that a user should not be able to learn anything, from either the content stored in its cache or the server transmissions, about a file it did not request. We propose a feasible scheme for this setting and demonstrate its order-optimality by deriving information-theoretic lower bounds.

I. INTRODUCTION

Broadband data consumption has grown at a rapid pace over last couple of decades, owing in great part to multimedia applications such as Video-on-Demand [1]. Content delivery networks attempt to mitigate this extra load on the communication network by deploying storage units or caches where some of the popular content can be pre-fetched during the off-peak hours.

Content caching and delivery has been studied extensively in the literature, see for example [16], [3], [2] and references therein. However, most of the work proposes caching schemes where those parts of the requested files that are available at nearby caches are served locally and the remaining parts are served by a remote server via separate unicast transmissions to the users. In contrast, recent work [8], [9] has studied an information-theoretic formulation of the problem and proposed the idea of *coded caching* which uses the available cache memory to not only provide local access to content but to also generate coded-multicasting opportunities among users with different demands. The setup studied in [8], [9] consists of a server communicating to a set of users, each equipped with a cache of uniform size, over a broadcast link and the objective

is to minimize the worst-case server transmission rate, over all feasible user demands. For this setup, coded caching is shown to provide significant benefits over traditional caching and delivery, and is in fact within a constant factor of the optimal. In this work, we consider a similar problem setup where we have the additional constraint that no user should be able to obtain any information, from its cache content as well as the server transmission, about any file other than the one it has requested. We call this setup ‘*secretive*’ and devise a *secretive coded caching* scheme for the setup based on ideas from secret sharing [4], [13], and show that the performance of this scheme is within a constant factor of the optimal for almost all parameter values of interest. We also show that using secret sharing, we can arrive at a decentralized scheme, following the setting of [9], wherein caching is carried out independently at each user. This mode of caching allows for varying number of users and for the absence of a centralized coordinating server. The performance of this decentralized scheme is again within a constant factor of the optimal.

The results of [8], [9] on coded caching have been extended in several other directions as well, ranging from heterogeneous cache sizes [15], unequal file sizes [17], to improved converse arguments [5], [12]. Content caching and delivery has also been studied in the context of device-to-device networks, multi-server topologies, and heterogeneous wireless networks in [7], [14], and [6] respectively. The work closest to ours is [11], which considers the problem of *secure coded caching*, where the goal is to protect information about the files from an eavesdropper which can listen to the server transmissions. However, [11] doesn’t capture the notion of ‘*secrecy*’ that we consider here. Throughout the paper by ‘*security*’, ‘*secure*’ we mean protection against an eavesdropper and ‘*secretive*’, ‘*secrecy*’ refer to our problem of interest. While most of the work in this paper focuses on secretive coded caching, we will also briefly discuss the case where one requires both secrecy and security.

The rest of the paper is organized as follows. We describe the problem setup in Section II and present our main results in Section III. We discuss some examples in Section IV before describing our proposed secretive coded caching scheme in Section V and presenting converse arguments in Section VI. Order-optimality of the proposed scheme is established in Section VII. A decentralized variant of the proposed scheme is presented in Section VIII and we conclude with a discussion of our results in Section IX. A part of this work was presented in [10]; this manuscript has the complete proofs of all the results as well as original technical content, in particular Section VIII is new.

II. PROBLEM FORMULATION

Notation: For $n \in \mathbb{N}$, we denote by $[n]$ the set $\{1, 2, \dots, n\}$. A vector of random variables will be denoted by bold-faced upper case letters, e.g., $\mathbf{Y} = (Y_1, Y_2, \dots, Y_n)$. For a set $A \subseteq [n]$, we will denote the vector of random variables indexed by elements in A by \mathbf{Y}_A . Specifically, if $A = \{i_1, i_2, \dots, i_m\}$ where $1 \leq i_1 < i_2 < \dots < i_m \leq n$, we denote $\mathbf{Y}_A = (Y_{i_1}, Y_{i_2}, \dots, Y_{i_m})$.

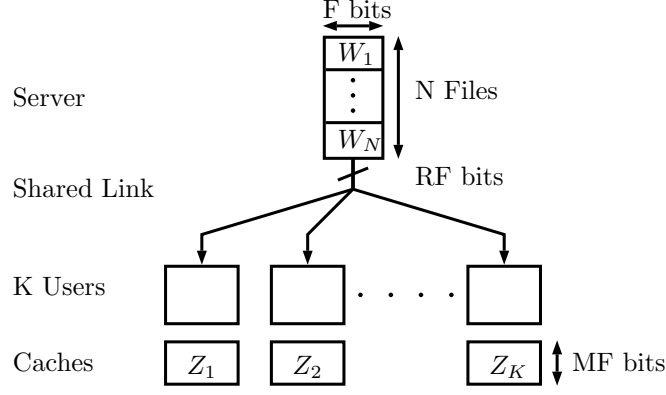


Fig. 1. In the setup above, a server containing N files, each of F bits, is connected via an error-free shared link to K users, each with a cache memory of size MF bits. The server multicasts through this link at a rate of at most RF bits.

We consider a single-hop content delivery network, as illustrated in Figure 1. The system consists of a server hosting a collection of N files, $\mathbf{W} = (W_1, W_2, \dots, W_N)$, each of size F bits. We will assume that W_1, W_2, \dots, W_N are independent random variables each distributed uniformly over $[2^F]$. The server is connected via a shared, error-free link to K users, each with a cache memory of size MF bits. We will refer to M as the normalized cache memory size.

The system works in two phases: a *placement phase* followed by a *delivery phase*. In the placement phase, the user caches are populated with content related to the N files using a possibly randomized scheme. Formally, we denote the content stored in cache k by a random variable Z_k which takes values in $[2^{MF}]$. The vector $\mathbf{Z} = (Z_1, Z_2, \dots, Z_K)$ is jointly distributed according to some conditional distribution $p_{\mathbf{Z}|\mathbf{W}}$. Note that the placement phase is performed without any prior knowledge of future user demands. During the delivery phase, each user requests one of the N files. The resulting demand vector $\mathbf{d} = (d_1, d_2, \dots, d_K)$ is revealed to all the users and the server. The server transmits a message $X_{\mathbf{d}}(\mathbf{W}, \mathbf{Z})$ of size RF bits on the shared link to the users.

Each user k generates an estimate \widehat{W}_{d_k} of its requested file W_{d_k} using only its stored cache content Z_k and the server transmission $X_{\mathbf{d}}(\mathbf{W}, \mathbf{Z})$. The probability of error of a placement and delivery scheme is given by

$$P_e \triangleq \max_{\mathbf{d} \in [N]^K} \mathbb{P}((\widehat{W}_{d_1}, \dots, \widehat{W}_{d_K}) \neq (W_{d_1}, \dots, W_{d_K})), \quad (1)$$

where the probability is over the files and the randomization in the placement phase, i.e., over the distribution $p_{\mathbf{W}\mathbf{Z}|\mathbf{W}}$. In addition to recovering the demanded files, we also want each user to not obtain any information about the other files. The information leakage of a placement and delivery scheme is defined as:

$$L \triangleq \max_{\mathbf{d} \in [N]^K} \max_{k \in [K]} I(\mathbf{W}_{[N]/\{d_k\}}; X_{\mathbf{d}}(\mathbf{W}, \mathbf{Z}), Z_k). \quad (2)$$

A placement and delivery scheme is said to be an (ϵ, δ) -*secretive scheme* if its probability of error $P_e \leq \epsilon$ and information leakage $L \leq \delta$.

The memory-rate pair (M, R) is said to be *secretively achievable*, if for any $\epsilon, \delta > 0$ and large enough file size F , there exists an (ϵ, δ) -secretive scheme. The object of interest in this paper is the optimal server transmission rate $R_S^*(M)$ for normalized cache memory size M , given by

$$R_S^*(M) \triangleq \inf\{R : (M, R) \text{ is secretively achievable}\}. \quad (3)$$

III. MAIN RESULTS

The main result of this paper is an approximate characterization of the optimal server transmission rate $R_S^*(M)$ for any normalized cache memory size M . We propose a secretive caching and delivery scheme to show the following upper bound on $R_S^*(M)$.

Theorem 1. For $M = \frac{Nt}{K-t} + 1$ with $t \in \{0, 1, \dots, K-2\}$, the following rate is secretively achievable

$$R_C(M) \triangleq \frac{K(N+M-1)}{N+(K+1)(M-1)}. \quad (4)$$

For $M = N(K-1)$, we achieve the rate $R_C(M) = 1$. Further, for any general $1 \leq M \leq N(K-1)$, the convex envelope of these points is achievable.

Some comments are in order. Note that the achievable rate $R_C(M) = 1$ for $M = N(K-1)$. This is in fact the minimum achievable rate for any secretive caching and delivery scheme, i.e. $R_S^*(M) \geq 1$, $\forall M$. Intuitively, this is because the information leakage as defined in (2) is constrained to be negligible for any secretive scheme, and this implies that the contents of a cache cannot provide any information about the requested file on its own. Hence, for a user to learn the file it requested, it must receive from the server at least F bits. We provide a formal proof in Section VI. Similarly, note that we only consider $M \geq 1$ in the above result. As we prove in Section VI, this is indeed a necessary condition for the existence of a secretive caching and delivery scheme.

The next result provides an information-theoretic lower bound on the server transmission rate of any secretive caching and delivery scheme.

Theorem 2. For $1 \leq M \leq N(K-1)$,

$$R_S^*(M) \geq \max_{s \in \{1, 2, \dots, \min\{N/2, K\}\}} \frac{s \lfloor N/s \rfloor - 1 - (s-1)M}{\lfloor N/s \rfloor - 1}. \quad (5)$$

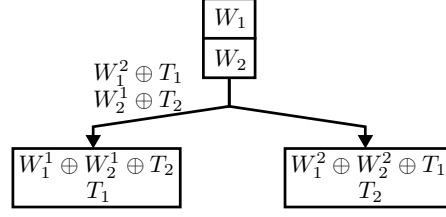


Fig. 2. Optimal scheme for $N = K = 2$, $M = 1$ achieving rate $R = 1$. Server transmission for the demand vector $(d_1, d_2) = (1, 2)$ is shown.

The above result is obtained using cut-set based arguments and is presented in section VI. The lower bound can be further improved by using non-cut set based arguments as shown in Section IX. However, the cut-set lower bound is indeed tight for the case with $N = K = 2$, as shown in Section IV. This lower bound also suffices to show that, in general, the server transmission rate of the proposed scheme is within a constant factor of the optimal for most regimes of interest:

Theorem 3. For $M \geq M_0 \triangleq 1 + \max \left\{ 0, \frac{N(K-N)}{(N-1)K+N} \right\}$,

$$1 \leq \frac{R_S^*(M)}{R_C(M)} \leq c_1, \quad (6)$$

where c_1 is a constant independent of all the system parameters.

It is easy to verify that $M_0 = 1$ for $N \geq K$. Recall that $M \geq 1$ is necessary for any secretive caching and delivery scheme and thus our proposed scheme is order-optimal for all permissible values of the normalised cache memory size M . For $N < K$, we have $M_0 \leq 1 + N/(N-1) < 5/2$ and thus the above result establishes the order-optimality of our proposed scheme for all regimes of interest except for $1 \leq M \leq 5/2$. This is because for $N < K$, the lower bound from theorem 5 depends only the number of files N . However, we expect the optimal rate to increase with the number of users K , since we have to ensure secrecy for a larger set of users.

IV. EXAMPLES

A. Optimal Scheme for $N = K = 2$ and $M = 1$

Figure 2 shows an example setup with $N = 2$ files and $K = 2$ users with normalized cache memory size $M = 1$. Partition the two files W_1, W_2 into two equal parts W_1^1, W_1^2 and W_2^1, W_2^2 respectively. Two independent and uniformly distributed *random keys*, T_1 and T_2 each of size $F/2$ bits, are generated. During the placement phase, the random keys and their combination with the file parts are put in the caches as shown in Figure 2. During the delivery phase, if the demand vector is $(d_1, d_2) = (1, 2)$, the server transmits $W_1^2 \oplus T_1$ and $W_2^1 \oplus T_2$, of total size F bits. It can be easily verified that both the users can recover

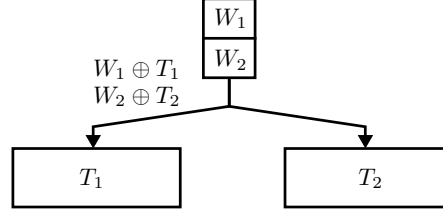


Fig. 3. Alternate scheme for $N = K = 2$, $M = 1$ achieving rate $R = 2$. Shown is the server transmission for demand vector $(d_1, d_2) = (1, 2)$.

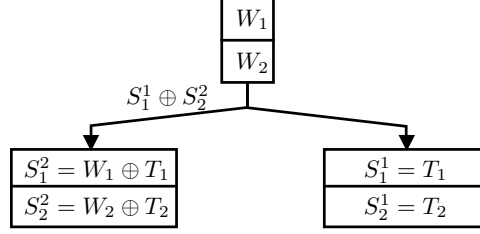


Fig. 4. Alternate scheme for $N = K = 2$, $M = 2$ achieving rate $R = 1$. Server transmission for the demand vector $(d_1, d_2) = (1, 2)$ is shown.

their requested files using their respective cache contents and the server transmission. Furthermore, neither user can derive any information about the file they did not request. Similarly, any other demand vector can also be secretly satisfied using a server transmission of size F bits. Specifically, note that when the users demand the same file, the server may send it in the clear. Thus, the memory-rate pair $(M = 1, R = 1)$ is secretly achievable. As mentioned before, $M \geq 1, R \geq 1$ are necessary conditions for feasibility in our setup and so the scheme presented above is in fact optimal.

While the scheme described above is optimal, it is not immediately clear how to generalize it to larger number of files and caches. Instead, below we discuss a sub-optimal scheme at two different memory-rate points. This scheme easily generalizes to our order-optimal scheme.

B. General Scheme at $M = 1$

At $M = 1$, we cache independent keys T_i of size F bits at each user $i \in [K]$, see Figure 3 for an illustration when $N = K = 2$. During delivery, the server transmits $W_{d_i} \oplus T_i$ for each user $i \in [K]$, resulting in a rate of K . It is easy to verify that each user is able to recover its requested file and obtains no information about the other files. Finally, note that the rate of this scheme matches the value of $R_C(1)$ (corresponding to $M = 1$) in (4).

C. General Scheme at $M = N(K - 1)$

At the other extreme when $M = N(K - 1)$, we use a *secret sharing scheme* as defined below:

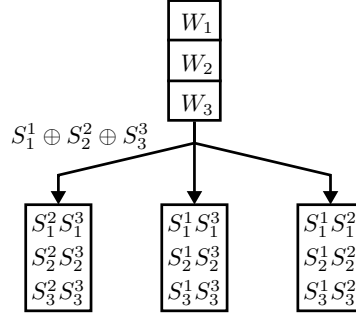


Fig. 5. Alternate scheme for $N = K = 3$, $M = 4$ achieving rate $R = 1$. Server transmission for the demand vector $(d_1, d_2, d_3) = (1, 2, 3)$ is shown.

Definition For $m < n$, by an (m, n) secret sharing scheme, we mean a “scheme” $p_{S_1, \dots, S_n|W}$ to generate n equal-sized shares S_1, \dots, S_n of a uniformly distributed secret W such that any m shares do not reveal any information about the secret and access to *all* the n shares completely reveals the secret. i.e.,

$$I(W; S_A) = 0, \forall A \subset [n] \text{ s.t. } |A| = m,$$

$$H(W|S_{[n]}) = 0.$$

It is easy to see that, for such a scheme, the shares must have a size of at least $\frac{\log |\mathcal{W}|}{n-m}$ bits, where \mathcal{W} is the alphabet of the secret. When $|\mathcal{W}|$ is large enough, secret sharing schemes which achieve this bound exist [4].

For each file W_i , $i \in [N]$, we use a $(K-1, K)$ secret sharing scheme, which provides K shares, each of size F bits and denoted by $\{S_i^j\}_{j=1}^K$, with the following properties:

- (i) No collection of $K-1$ shares reveals any information about the file W_i , and
- (ii) the file W_i can be recovered from its K shares $\{S_i^j\}_{j=1}^K$.

Figure 4 illustrates the case of $N = K = 2$, where the $K = 2$ shares for each file W_i are given by $S_i^1 = W_i \oplus T_i$ and $S_i^2 = T_i$, where T_i is a random key of size F bits. During the placement phase, different shares are stored in the various caches as follows: the contents of cache $k \in [K]$ is given by $Z_k = \{S_i^j : i \in [N], j \in [K], j \neq k\}$. Note that there are $N(K-1)$ shares stored in every cache, each of size F bits, and this agrees with the normalized cache memory size $M = N(K-1)$. Next, during the delivery phase, each user requests a file and the server transmits $\oplus_{k \in [K]} S_{d_k}^k$ of size F bits, resulting in a rate of 1. See Figure 5 for an illustration when $N = K = 3$. Since each user $k \in [K]$ already has all the shares $\{S_i^j\}_{i \in [N], j \neq k}$, the missing share of the demanded file $S_{d_k}^k$ can be obtained, and the file W_{d_k} can be reconstructed. Furthermore for any other file than the one requested, each user $k \in [K]$ only has $(K-1)$ shares which do not reveal any information because of the properties of the $(K-1, K)$ secret sharing scheme. Again, note that the rate of the proposed scheme agrees with the value of $R_C(M = N(K-1))$ in (4).

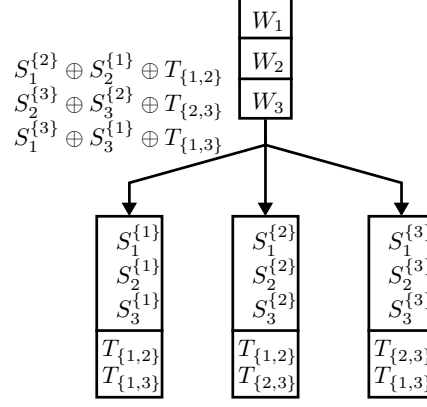


Fig. 6. General scheme for $N = K = 3$ with $t = 1, M = Nt/(K - t) + 1 = 5/2$ achieving rate $R = 3/2$. Server transmission for the demand vector $(d_1, d_2, d_3) = (1, 2, 3)$ is shown.

V. GENERAL ACHIEVABILITY SCHEME

In this section, we will generalize the ideas presented above to obtain a secretive caching and delivery scheme for all problem parameters N, K , and M , and characterize its rate to complete the proof of Theorem 1. In fact, we will propose an $(\epsilon = 0, \delta = 0)$ -secretive scheme, i.e. the probability of error as well as the information leakage are both zero.

We have already discussed the schemes which achieve $R_C(M)$ as defined in (4) at $M = 1$ and $M = N(K - 1)$. Next, we consider $M = Nt/(K - t) + 1$ for some $t \in \{1, \dots, K - 2\}$. We use a $((\binom{K-1}{t-1}, \binom{K}{t}))$ secret sharing scheme to create $\binom{K}{t}$ shares, each of size $F_s = \frac{F}{\binom{K}{t} - \binom{K-1}{t-1}} = \frac{Ft}{(K-t)\binom{K-1}{t-1}}$ bits, for each file $W_i, i \in [N]$. For each file W_i , we denote its shares by $\mathbf{D}_i \triangleq \{S_i^L : L \subset [K], |L| = t\}$ and define $\mathbf{C}_i^k \triangleq \{S_i^L : L \subset [K], |L| = t, k \in L\}$. Then for any $k \in [K]$, the shares satisfy the following properties:

$$I(W_{[N]}; \bigcup_{i \in [N]} \mathbf{C}_i^k) = 0, \quad (7)$$

$$I(W_{[N] \setminus \{d_k\}}; \bigcup_{i \in [N]} \mathbf{C}_i^k \cup \mathbf{D}_{d_k}) = 0, \quad (8)$$

$$H(W_{d_k} | \mathbf{D}_{d_k}) = 0. \quad (9)$$

The identities (7), (8) imply that $\binom{K-1}{t-1}$ shares of a file reveal no information about it and shares of one file do not provide information about another file since they are independent; and (9) implies that $\binom{K}{t}$ shares of a file are sufficient for recovering it without error. During the placement phase, share S_i^L is placed in the cache of user k if $k \in L$. Thus $\bigcup_{i \in [N]} \mathbf{C}_i^k$ precisely denotes the shares cached at user k . Since we have $\binom{K-1}{t-1}$ shares of each of the N files in every user cache, the total memory size in bits needed for storing the shares is given by

$$F_s \cdot N \cdot \binom{K-1}{t-1} = \frac{Ft}{(K-t)\binom{K-1}{t-1}} \cdot N \cdot \binom{K-1}{t-1} = \frac{Nt \cdot F}{K-t}. \quad (10)$$

In addition to the shares, for each subset $V \subset [K]$ of users of size $|V| = t + 1$, an independently and uniformly generated key T_V of size F_s bits is cached at each user $k \in V$. For each user, the cache memory in bits needed to store the keys is given by

$$F_s \binom{K-1}{t} = \frac{Ft}{(K-t)\binom{K-1}{t-1}} \cdot \binom{K-1}{t} = F. \quad (11)$$

Combining (10) and (11), the total memory needed per cache is given by $(\frac{Nt}{K-t} + 1)F$ bits which agrees with $M = Nt/(K-t) + 1$. See Figure 6 for an illustration of the placement phase when $N = K = 3$ and $t = 1$.

During the delivery phase, the demand vector (d_1, \dots, d_K) is revealed to the server and the users. Then for each $V \subset [K]$ such that $|V| = t + 1$, the server transmits $T_V \oplus_{k \in V} S_{d_k}^{V \setminus \{k\}}$ on the shared link to the users. See Figure 6 for an example. Consider one such subset V and its associated server transmission. From the placement phase, each $k \in V$ has the key T_V as well as all the shares in the message except $S_{d_k}^{V \setminus \{k\}}$, and hence each user k can recover the share $S_{d_k}^{V \setminus \{k\}}$. It is easy to verify that at the end of the delivery phase, each user k would possess all the $\binom{K}{t}$ shares of its requested file W_{d_k} and thus, from (9), can recover it without error. Furthermore, the scheme ensures that the server transmissions do not reveal any information to a user about files it did not request. This combined with (7), (8) ensures that the information leakage, as defined in (2), of the placement and delivery phases of the proposed scheme is zero. Thus, we have a secretive caching and delivery scheme. Finally, the server transmission size in bits of our proposed scheme at $M = Nt/(K-t) + 1$ is given by

$$\binom{K}{t+1} \cdot F_s = \frac{\binom{K}{t} t \cdot F}{(K-t)\binom{K-1}{t-1}} = \frac{KF}{1+t}.$$

Substituting $t = (M-1)K/(N+M-1)$, we obtain the achievable rate expression $R_C(M)$ as defined in Theorem 1.

VI. A LOWER BOUND

In this section, we provide a lower bound on the optimal server transmission rate $R_S^*(M)$, as defined in (3). Our proof is along similar lines as [8], [11].

Consider a tuple (M, R) which is secretly achievable. Fix $s \in [\min\{N/2, K\}]$ and consider users $1, 2, \dots, s$. Suppose user i requests file i . Since the tuple (M, R) is secretly achievable, for any $\epsilon > 0$, there exists a secretive placement and delivery scheme such that each user i can recover its requested file with a server transmission of rate R along with its cache content Z_i with probability of error at most ϵ . Furthermore for any $\delta > 0$, the information leakage to any user about a file other than the one it had requested is at most δ . Next, consider another scenario where each user $i \in [s]$ requests file $s+i$. Again, since the tuple (R, M) is secretly achievable, the recovery and secrecy conditions still hold true.

One can repeat the same argument for $\lfloor N/s \rfloor$ different request patterns. Let X_l denote the server transmission corresponding to the l^{th} request instance when the demand pattern is given by $(d_1 = (l-1)s + 1, d_2 = (l-1)s + 2, \dots, ls)$. Define $\mathbf{X}_{\lfloor N/s \rfloor} \triangleq (X_i : i \in [\lfloor N/s \rfloor])$ and recall that $\mathbf{Z}_{[s]} = (Z_i : i \in [s])$, $\mathbf{W}_{\lfloor N/s \rfloor} = (W_i : i \in [\lfloor N/s \rfloor])$. Also, let $\widehat{\mathbf{W}} \triangleq$

$\mathbf{W}_{[s\lfloor N/s \rfloor] \setminus \{(l-1)s+k\}}$, $\widehat{\mathbf{X}} \triangleq \mathbf{X}_{[\lfloor N/s \rfloor] \setminus \{l\}}$ and $\widehat{\mathbf{Z}} \triangleq \mathbf{Z}_{[s] \setminus \{k\}}$. In words, $\widehat{\mathbf{W}}$ denotes the vector of all files except the one requested by the user k in the l^{th} request instance, $\widehat{\mathbf{X}}$ denotes the vector of server transmissions in all the request instances except the l^{th} one, and $\widehat{\mathbf{Z}}$ refers to the contents of all the user caches except the k^{th} one.

Then, from the recovery and secrecy conditions, for (M, R) to be secretively achievable, for $l \in [\lfloor N/s \rfloor]$, $k \in [s]$, we have using Fano's inequality in (1) and using (2)

$$H(\mathbf{W}_{[s\lfloor N/s \rfloor]} | \mathbf{X}_{[\lfloor N/s \rfloor]}, \mathbf{Z}_{[s]}) \leq H_b(\epsilon) + \epsilon NF \quad (12)$$

$$I(\widehat{\mathbf{W}}; X_l, Z_k) \leq \delta. \quad (13)$$

Where for any $x \in [0, 1]$, $H_b(x)$ is the binary entropy function. Then, we have

$$\begin{aligned} (s\lfloor N/s \rfloor - 1)F &= H(\widehat{\mathbf{W}}) \\ &= I(\widehat{\mathbf{W}}; \mathbf{X}_{[\lfloor N/s \rfloor]}, \mathbf{Z}_{[s]}) + H(\widehat{\mathbf{W}} | \mathbf{X}_{[\lfloor N/s \rfloor]}, \mathbf{Z}_{[s]}) \\ &\stackrel{(a)}{\leq} I(\widehat{\mathbf{W}}; \mathbf{X}_{[\lfloor N/s \rfloor]}, \mathbf{Z}_{[s]}) + H_b(\epsilon) + \epsilon NF \\ &= I(\widehat{\mathbf{W}}; X_l, Z_k) + I(\widehat{\mathbf{W}}; \widehat{\mathbf{X}}, \widehat{\mathbf{Z}} | X_l, Z_k) + H_b(\epsilon) + \epsilon NF \\ &\stackrel{(b)}{\leq} I(\widehat{\mathbf{W}}; \widehat{\mathbf{X}}, \widehat{\mathbf{Z}} | X_l, Z_k) + H_b(\epsilon) + \epsilon NF + \delta \\ &\leq H(\widehat{\mathbf{X}}, \widehat{\mathbf{Z}}) + H_b(\epsilon) + \epsilon NF + \delta \\ &\leq \sum_{i=1, i \neq l}^{\lfloor N/s \rfloor} H(X_i) + \sum_{j=1, j \neq k}^s H(Z_j) + H_b(\epsilon) + \epsilon NF + \delta \\ &\leq (\lfloor N/s \rfloor - 1)R_S^*(M)F + (s-1)MF + H_b(\epsilon) + \epsilon NF + \delta \end{aligned}$$

where (a), (b) follow from (12), (13) respectively. Rearranging the terms, we get

$$R_S^*(M) \geq \frac{s\lfloor N/s \rfloor - 1 - (s-1)M - (H_b(\epsilon) + \epsilon NF + \delta)/F}{\lfloor N/s \rfloor - 1}.$$

The statement of Theorem 2 then follows by noting that the above inequality holds true for any $s \in \{1, 2, \dots, \min\{N/2, K\}\}$ and by choosing ϵ, δ to be arbitrarily small.

VII. ORDER-OPTIMALITY

In this section, we show that the achievable rate $R_C(M)$ is within a constant factor with the above information theoretic lower bound. In particular, we prove that for $M \geq 1 + \max\{\frac{N(K-N)}{(N-1)K+N}, 0\}$,

$$\frac{R_C(M)}{R_S^*(M)} \leq 16.$$

Define $M_S \triangleq M - 1$. Then using $N/s - 1 \leq \lfloor N/s \rfloor$ and $M_S = M - 1$ in Theorem 2, we have

$$R_S^*(M) \geq \max_{s \in \{1, 2, \dots, \min\{N, K\}\}} s - M_S \frac{s(s-1)}{N-2s}. \quad (14)$$

Also, the achievable rate expression $R_C(M)$ may be written as

$$R_C(M) = \frac{K}{1 + KM_S/(N + M_S)}.$$

For values of M in our range of interest, if $K \leq N$,

$$R_C(M) = \frac{K}{1 + KM_S/(N + M_S)} \leq K.$$

On the other hand, if $K > N$,

$$R_C(M) \leq R_C \left(\frac{N(K-N)}{(K+1)N-K} \right) = N.$$

Thus, we have $R_C(M) \leq \min\{N, K\}$.

Case 1: $\min\{N, K\} \leq 16$.

In this case,

$$R_C(M) \leq \min\{N, K\} \leq 16.$$

And since $R_S^*(M) \geq 1$,

$$\frac{R_C(M)}{R_S^*(M)} \leq 16.$$

Case 2: $\min\{N, K\} > 16$.

In this case we consider 3 regions based on the values of M_S .

Region I: $0 \leq M_S < \max\{N, K\}/(K-1)$.

Let $s = \lfloor 0.205 \min\{N, K\} \rfloor^1$ in (14), using which we obtain

$$\begin{aligned} R_S^*(M) &\geq s - M_S \frac{s(s-1)}{N-2s} \\ &\geq \lfloor 0.205 \min\{N, K\} \rfloor - M_S \frac{\lfloor 0.205 \min\{N, K\} \rfloor (\lfloor 0.205 \min\{N, K\} \rfloor - 1)}{N - 2\lfloor 0.205 \min\{N, K\} \rfloor} \\ &\geq \min\{N, K\} \left(0.205 - \frac{1}{\min\{N, K\}} - \frac{0.205(0.205 \min\{N, K\} - 1)(\frac{\max\{N, K\}}{K-1})}{N - 2 * 0.205 \min\{N, K\}} \right). \end{aligned} \quad (15)$$

Now consider the expression $\max\{N, K\}(0.205 \min\{N, K\} - 1)/(K-1)$. If $N < K$,

$$\begin{aligned} \max\{N, K\} \frac{0.205 \min\{N, K\} - 1}{K-1} &= K \frac{0.205N - 1}{K-1} \\ &\leq 0.205N(16/15). \end{aligned}$$

¹ $s \geq 1$ for $\min\{N, K\} \geq 5$. This assumption is however, not critical for our analysis.

And if $N \geq K$,

$$\begin{aligned} \max\{N, K\} \frac{0.205 \min\{N, K\} - 1}{K - 1} &= N \frac{0.205K - 1}{K - 1} \\ &\leq 0.205N \\ &\leq 0.205N(16/15). \end{aligned}$$

Plugging this into (15) we get

$$\begin{aligned} R_S^*(M) &\geq \min\{N, K\} \left(0.205 - \frac{1}{\min\{N, K\}} - \frac{16}{15} \frac{0.205^2}{1 - 2 * 0.205 \frac{\min\{N, K\}}{N}} \right) \\ &\geq \min\{N, K\} \left(0.205 - \frac{1}{16} - \frac{16}{15} \frac{0.205^2}{1 - 2 * 0.205} \right) \\ &\geq \min\{N, K\} / 16. \end{aligned}$$

This gives

$$\frac{R_C(M)}{R_S^*(M)} \leq 16.$$

Region II: $\max\{N, K\} / (K - 1) \leq M_S < N/15$.

In this region, note that

$$1/M_S \geq (K - 1) / \max\{N, K\} \quad (16)$$

and that

$$M_S < N/15 \Rightarrow \frac{N}{N + M_S} \geq \frac{15}{16}. \quad (17)$$

Now,

$$\begin{aligned} R_C(M) &= \frac{K}{1 + KM_S / (N + M_S)} \\ &\leq \frac{K}{KM_S / (N + M_S)} \leq \frac{N + M_S}{M_S}. \end{aligned}$$

Letting $s = \lfloor 0.198 \frac{N + M_S}{M_S} \rfloor^2$ in (14) and following steps similar to the one used to get (15) we have

$$R_S^*(M) \geq \frac{N + M_S}{M_S} \left(0.198 - \frac{M_S}{N + M_S} - \frac{0.198^2}{N / (N + M_S) - 2 * 0.198 / M_S} \right).$$

Using (16) and (17) in the above inequality, we get

$$\begin{aligned} R_S^*(M) &\geq \frac{N + M_S}{M_S} \left(0.198 - \frac{1}{16} - \frac{0.198^2}{15/16 - 2 * 0.198} \right) \\ &\geq \frac{N + M_S}{M_S} \frac{1}{16}. \end{aligned}$$

²Since $M_S < N/15$ in this regime, $s \geq 1$ and similarly $\max\{N, K\} / (K - 1) \leq M_S$ guarantees that $s < \min\{N/2, K\}$

Hence,

$$\frac{R_C(M)}{R_S^*(M)} \leq 16.$$

Region III: $N/15 \leq M_S$.

Note that $N/15 \leq M_S \Rightarrow (N + M_S)/16 \leq M_S$, using which

$$R_C(M) = \frac{K}{1 + K \frac{M_S}{N + M_S}} \leq \frac{K}{1 + K \frac{1}{16}} \leq 16.$$

Using $R_S^*(M) \geq 1$ with the above inequality, we get

$$\frac{R_C(M)}{R_S^*(M)} \leq 16.$$

This proves Theorem 3.

VIII. DECENTRALIZED SCHEME

The *secretive* coded caching scheme discussed in the above sections requires prior knowledge of the number of users K during the placement phase. This is a disadvantage and our proposed scheme becomes inapplicable if users choose to leave or seek to join the system before the delivery phase. With this motivation, we briefly discuss a ‘decentralized’ variant of our earlier ‘centralized’ scheme, wherein the placement phase is carried out at each user independently of the other users. The scheme is inspired by the ideas presented in [9]. We further show that this scheme while being ‘decentralized’, is still order-optimal with respect to the information-theoretic lower bound.

In the placement phase, for each file $W_i, i \in [n]$, we generate $G \in \mathbb{N}$ shares, denoted by \mathbf{D}_i . These shares are such that the file W_i can be reconstructed using \mathbf{D}_i , but any subset of qG shares³ from \mathbf{D}_i provide no information about the file W_i , where

$$q \triangleq (M - 1)/(M + N - 1).$$

Let h denote the size of each share. By working with a large enough F , we can obtain a share size of (approximately) $h = F/(G - qG)$ bits and thus

$$Gh = F/(1 - q).$$

Now, for each user $k \in [K]$ and for each file $W_i, i \in [n]$, we independently pick a subset of qG shares from \mathbf{D}_i uniformly at random and cache them at user k . Next, we generate a collection of $G \cdot (1 - q)/q$ independent keys \mathbf{U} , where each key is of

³We ignore the integrality constraint in this section for ease of exposition. We can approximate the rates shown as closely as desired by choosing a large enough file size F .

size h bits. Then for each user $k \in [K]$, we independently pick a subset of $(1 - q)G$ keys from \mathbf{U} uniformly at random and cache them at user k . Thus, after placement, the total memory used at each cache in bits is given by

$$\begin{aligned}
qG \cdot h \cdot N + (1 - q)G \cdot h &= (qN + 1 - q) \cdot Gh \\
&= (qN + 1 - q) \cdot F/(1 - q) \\
&= (qN/(1 - q) + 1) \cdot F \\
&= \left(\frac{(M - 1)N/(M + N - 1)}{N/(M + N - 1)} + 1 \right) \cdot F \\
&= MF.
\end{aligned}$$

Thus, the memory constraint is satisfied at each cache.

For any subset of users $S \subseteq [K]$ and any file W_i , let \mathbf{G}_i^S denote the shares of file W_i that are cached⁴ exclusively at all the users in this subset S , i.e., these shares are cached at each user in S and not cached at any user in $[K] \setminus S$. It is easy to verify that $E[|\mathbf{G}_i^S|] = q^{|S|}(1 - q)^{(K - |S|)}G$. Similarly, let \mathbf{U}^S denote the set of keys cached exclusively in the subset S , then $E[|\mathbf{U}^S|] = q^{|S| - 1}(1 - q)^{(K - |S| + 1)}G$.

During the delivery phase, the users reveal the demand vector (d_1, d_2, \dots, d_K) to the server. For each non-empty subset $S \subseteq [K]$, provided the following condition is satisfied

$$|\mathbf{U}^S| \geq \max_{k \in S} |\mathbf{G}_{d_k}^{S \setminus k}|, \quad (18)$$

the server transmits a vector sum^5 of file shares encrypted with the corresponding key as $(\oplus_{k \in S} \mathbf{G}_{d_k}^{S \setminus k}) \oplus \mathbf{U}^S$. If (18) is indeed satisfied for all $S \subseteq [K]$ (as we will show in the sequel is the case with high probability), this allows each user $k \in S$ to recover $\mathbf{G}_{d_k}^{S \setminus k}$ since it has access to \mathbf{U}^S and $\mathbf{G}_{d_j}^{S \setminus j}$ for all $j \in S \setminus k$. Thus, from the various server transmissions, each user k gains access to all the shares of its requested file W_{d_k} which enables its reconstruction. The expected server transmission rate $E[R_D(M)]$ for this scheme can be bounded by going over all subsets $S \subseteq [K]$ ($S \neq \emptyset$) and is given by

$$\begin{aligned}
E[R_D(M)]F &\leq \sum_{j=1}^K \binom{K}{j} q^{j-1} (1 - q)^{K - (j-1)} Gh \\
&= Gh \frac{1 - q}{q} (1 - (1 - q)^K) \\
&= \frac{F(1 - (1 - q)^K)}{q} \quad (19)
\end{aligned}$$

⁴ \mathbf{G}_i^ϕ refers to those shares which are not cached at any user

⁵We zero pad the vectors $\mathbf{G}_{d_k}^{S \setminus k}$ before summing so that they are all of the same size, namely, $|\mathbf{U}^S|$.

where we used the fact that $E[|\mathbf{U}^S|] = q^{|S|-1}(1-q)^{(K-|S|+1)}G$ in the first step. Recall that $q = (M-1)/(M+N-1)$. Furthermore, by letting the file size F and the number of shares G grow large, we get that the server transmission rate $R_D(M)$ approaches $E[R_D(M)]$ by the law of large numbers.

We now argue that our secrecy condition (2) is satisfied. The placement phase provides a user with no more than qG shares of any file, so no information about other files is leaked. Recall that, in the delivery phase, for each subset $S \subseteq [K]$ ($S \neq \phi$), the server transmits an encrypted sum of shares $(\oplus_{k \in S} \mathbf{G}_{d_k}^{S \setminus k}) \oplus \mathbf{U}^S$. Since for each subset S , the server transmits an encrypted sum only if $|\mathbf{U}^S| \geq \max_{k \in S} |\mathbf{G}_{d_k}^{S \setminus k}|$ is satisfied, each user obtains access only to the shares of its requested file W_{d_k} in the delivery phase, and no new information about the shares of other files is revealed. This ensures that our secrecy condition (2) will be guaranteed.

It now only remains to show that (18) holds for all non-empty $S \subseteq [K]$ with probability approaching 1 as the file size F and the total number of shares G grow large. For the scheme described above, we have $E[|\mathbf{G}_{d_k}^{S \setminus k}|] = q^{|S|-1}(1-q)^{(K-|S|+1)}G = E[|\mathbf{U}^S|]$ for each $S \subseteq [K]$ and thus the desired inequality is not guaranteed with high probability. This can be handled by increasing the number of keys by a small amount. In particular, for some small $r > 0$, let $\hat{q} = (M-1)/(M+N-1+r)$ and the total number of available keys be $(r + (1-\hat{q})/\hat{q})G$. Then, for any subset S , we will have $E[|\mathbf{G}_{d_k}^{S \setminus k}|] = \hat{q}^{|S|-1}(1-\hat{q})^{(K-|S|+1)}G$ and $E[|\mathbf{U}^S|] = \hat{q}^{|S|}(1-\hat{q})^{(K-|S|)}((1-\hat{q})/\hat{q} + r)G > E[|\mathbf{G}_{d_k}^{S \setminus k}|]$. We can then use any appropriate concentration inequality (for example, Chebyshev's inequality) to show that the desired condition $|\mathbf{U}^S| \geq \max_{k \in S} |\mathbf{G}_{d_k}^{S \setminus k}|$ is satisfied with high probability for all subsets S (for a sufficiently large number of shares G which in turn calls for a sufficiently large file size F); we skip the details here for brevity. Since r can be chosen to be any small positive constant, the server transmission rate can be made arbitrarily close to the expression in (19). This completes the description of the decentralized version of our proposed secretive coded caching scheme.

In Figure 7, $R_D(M)$ is plotted against the centralized rate, $R_C(M)$ and the cut-set lower bound. In the plot we see that the decentralized rate is very 'close' to the centralized rate for a large range of memory sizes. Next, we show that for any $M \geq 1$, $R_D(M)/R_C(M) \leq 2$. We prove this by noting that $R_C(M)$ can be re-written as $R_C(M) = 1/(q(1 + \frac{1}{Kq}))$. Thus, we have $R_D(M)/R_C(M) = (1 + \frac{1}{Kq})(1 - (1-q)^K)$. Then, consider the following cases:

Case 1: $Kq \geq 1$. In this case

$$(1 + \frac{1}{Kq})(1 - (1-q)^K) \leq 1 + \frac{1}{Kq} \leq 2,$$

Case 2: $Kq < 1$. Here

$$(1 + \frac{1}{Kq})(1 - (1-q)^K) \leq (1 + \frac{1}{Kq})Kq = Kq + 1 \leq 2$$

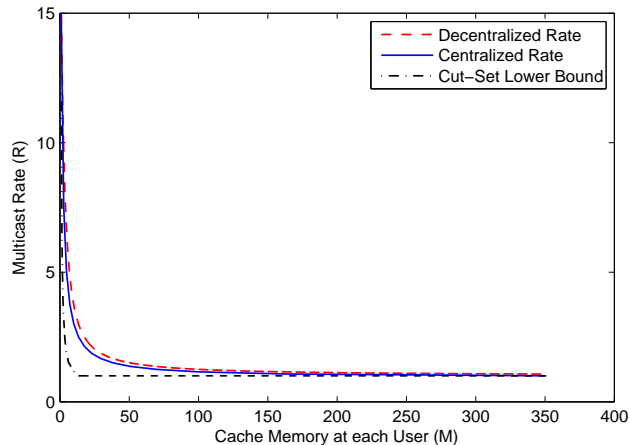


Fig. 7. The plot shows the achievable rates and the converse for a setup with $N = 25$ files and $K = 15$ users. The dashed red line is the decentralized rate, $R_D(M)$. The solid blue line is the rate obtained with the earlier centralized scheme, $R_C(M)$. The dash-dot black line is the cut-set lower bound..

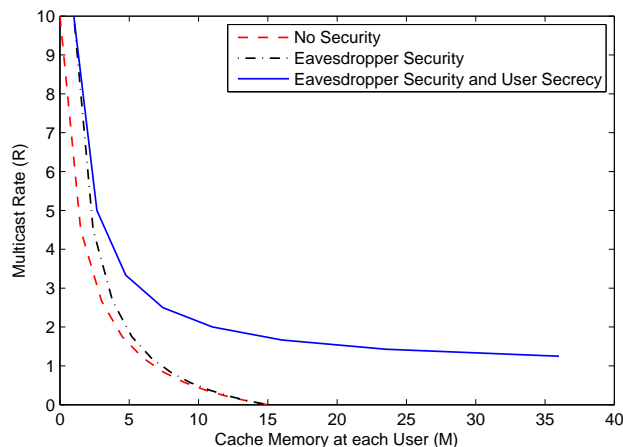


Fig. 8. The plot shows the achievable rates for a setup with $N = 15$ files and $K = 10$ users under various conditions. The dashed red line is the achievable rate with no security obtained in [8]. The dash-dot black line is the achievable rate with only eavesdropper security achieved in [11]. The solid blue line is our achievable rate $R_C(M)$ with eavesdropper security and secrecy against users.

and this proves our result. The result also establishes the order-optimality of our decentralized scheme.

IX. DISCUSSION

As mentioned before, the work closest to ours is [11], which studied the optimal server transmission rates needed to keep the files secure from an eavesdropper listening to the transmissions on the shared link. In contrast, we imposed the secrecy requirement that users should not be able to learn about files they did not request. An obvious scenario of interest is when both the conditions, security against an eavesdropper and secrecy from users, have to be satisfied. Let $R_{SE}^*(M)$ denote the optimal server transmission rate in such a setup, as a function of the normalized cache size M .

As an example, recall the setup in Figure 2 with $N = K = 2$ and $M = 1$ for which the minimum rate for a secretive scheme is given by $R_S^*(M = 1) = 1$. Under the optimal scheme illustrated in Figure 2, when both users request say file W_1 , the server simply transmits W_1 on the shared link. While this sufficed for satisfying the user secrecy constraints, clearly it will not work in the presence of an eavesdropper. In fact, the memory-rate tuple $(M = 1, R = 1)$ is not feasible if we insist on both secrecy from users and security against the eavesdropper. The optimal server transmission rate in this scenario is given by $R_{SE}^*(M) = 3 - M$ for $1 \leq M \leq 2$. A proof for this is provided in Appendix A.

While the optimal scheme in the above example did not protect against eavesdroppers, the general achievability scheme proposed in Section V does in fact have this additional property since each server transmission to a subset V of users is protected using a key⁶ T_V . Thus, an eavesdropper who has access to these transmissions can obtain no information about the files. This implies that the rate function $R_C(M)$ as defined in (4) is in fact achievable for the setup with both security and secrecy constraints, i.e. $R_{SE}^*(M) \leq R_C(M)$. Furthermore, it is easy to see that the lower bounds in Theorem 2 and the order-optimality result in Theorem 3 also continue to hold. Thus, the transmission rate for our proposed scheme is still within a constant factor of the optimal when both security and secrecy conditions are imposed.

Figure 8 plots the order-optimal transmission rates under various constraints. Note that when either no constraint or only the security against eavesdropper constraint is imposed, the achievable rate is zero at $M = N$. On the other hand, once the user secrecy condition is activated, the minimum achievable rate for any value of M is one. Furthermore, as the figure illustrates, the gap between the rate with no security and the rate with security against an eavesdropper is not very large. This was in fact shown to be at most a constant factor in [11]. The same continues to hold for a large memory regime, $1 < M < N \frac{K-1}{2K}$, when a further user secrecy constraint is also added.

Now, for the case with eavesdropper security alone, a decentralized scheme was presented in [11]. However, the scheme presented therein is not strictly decentralized since the key placement involved knowing the total number of users.

On the other hand, our scheme elucidated in Section VIII is in fact, strictly decentralized since the key placement is carried out independent of the other users. It turns out that the key placement for the eavesdropper setup in [11] can be made strictly decentralized by independently caching at each user, a subset of size $q_1 \cdot (1 - q_1) / q_1 \cdot F$ bits chosen uniformly at random from a key stream of size $(1 - q_1) / q_1 \cdot F$, where $q_1 \triangleq (M - 1) / (N - 1)$. The file placement and delivery procedures are identical to [11]. As a result, we obtain the same transmission rate as [11], but via a strictly decentralized scheme.

⁶Strictly speaking, this is not true for the scheme at the extreme memory point $M = N(K - 1)$ since we don't use a key to protect the server transmission. However, this can be easily fixed without affecting order-optimality by additionally storing a common key in each cache and securing the server transmission with this key.

APPENDIX A

Here we prove the claim made in section IX, that for case where $N = K = 2$ with eavesdropper security and secrecy constraints, the optimal transmission rate is given by $R_{SE}^*(M) = 3 - M$.

We first lower bound the optimal rate $R_{SE}^*(M)$. Consider two demand vectors $(d_1, d_2) = (1, 2)$ and $(d_1, d_2) = (1, 1)$ and let the corresponding deliveries be denoted as X_1 and X_2 . Denoting the files (W_1, W_2) as (A, B) , the demanded files are simply (A, B) , (A, A) respectively. We thus have

$$\begin{aligned}
R_{SE}^*(M)F + MF &\geq H(X_1) + H(Z_2) \\
&\geq H(X_1|Z_1) + H(Z_2|X_2) \\
&= I(X_1; A|Z_1) + H(X_1|A, Z_1) + I(Z_2; A|X_2) + H(Z_2|A, X_2) \\
&\geq I(X_1; A|Z_1) + I(Z_2; A|X_2) + H(X_1, Z_2|A, X_2, Z_1) \\
&= I(X_1; A|Z_1) + I(Z_2; A|X_2) + I(B; X_1, Z_2|A, X_2, Z_1) + H(X_1, Z_2|B, A, X_2, Z_1) \\
&\geq I(X_1; A|Z_1) + I(Z_2; A|X_2) + I(B; X_1, Z_2|X_2, Z_1) \\
&\geq 3F - 3\epsilon F
\end{aligned}$$

Note that $I(X_1; A|Z_1) = I(A; X_1, Z_1)$, $I(Z_2; A|X_2) = I(Z_2, X_2; A)$ and $I(B; X_1, Z_2|X_2, Z_1) = I(B; X_1, Z_2, X_2, Z_1)$ and thus the last step follows from Fano's inequality. A point to note here is that $I(Z_2; A|X_2) = I(Z_2, X_2; A)$ only because the eavesdropper security constraint was imposed. In the absence of this, X_2 and A are no longer independent and the proof fails.

For achievability, a minor addition to the coded placement scheme of Figure 2, that is additionally caching a common key K of F bits at both users allows us to achieve the point $(M, R) = (2, 1)$. Further, as we had already achieved the point $(M, R) = (1, 2)$, as shown in Figure 3, by memory sharing we're able to achieve the lower bound $R_{SE}^*(M) \geq 3 - M$, giving us a complete characterization of the rate region.

REFERENCES

- [1] "Cisco visual networking index (vni) global mobile data traffic forecast update," 2013, http://www.gsma.com/spectrum/wp-content/uploads/2013/03/Cisco_VNI-global-mobile-data-traffic-forecast-update.pdf.
- [2] S. Borst, V. Gupta, and A. Walid, "Distributed caching algorithms for content distribution networks," in *Proc. IEEE INFOCOM*, Mar. 2010, pp. 1478–1486.
- [3] L. Breslau, P. Cao, L. Fan, G. Phillips, and S. Shenker, "Web caching and Zipf-like distributions: Evidence and implications," in *Proc. IEEE INFOCOM*, Mar. 1999, pp. 126–134.
- [4] R. Cramer, I. B. Damgård, and J. B. Nielsen, *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.
- [5] H. Ghasemi and A. Ramamoorthy, "Improved lower bounds for coded caching," *arXiv:1501.06003 [cs.IT]*, 2015.

- [6] N. Golrezaei, K. Shanmugam, A. G. Dimakis, A. F. Molisch, and G. Caire, "Femtocaching: Wireless video content delivery through distributed caching helpers," in *Proc. IEEE INFOCOM*, Mar. 2012, pp. 1107–1115.
- [7] M. Ji, G. Caire, and A. F. Molisch, "Wireless device-to-device caching networks: Basic principles and system performance," *arXiv:1305.5216 [cs.IT]*, 2013.
- [8] M. A. Maddah-Ali and U. Niesen, "Fundamental limits of caching," *IEEE Trans. Inf. Theory*, vol. 60, no. 5, pp. 2856–2867, May 2014.
- [9] —, "Decentralized coded caching attains order-optimal memory-rate tradeoff," *IEEE/ACM Trans. Netw.*, pp. 1029–1040, Aug. 2015.
- [10] V. Ravindrakumar, P. Panda, N. Karamchandani, and V. Prabhakaran, "Fundamental limits of secretive coded caching," in *Proc. IEEE ISIT*, 2016, pp. 425–429.
- [11] A. Sengupta, R. Tandon, and T. C. Clancy, "Fundamental limits of caching with secure delivery," *Information Forensics and Security, IEEE Transactions on*, vol. 10, no. 2, pp. 355–370, 2015.
- [12] —, "Improved approximation of storage-rate tradeoff for caching via new outer bounds," in *Proc. IEEE ISIT*, June 2015, pp. 1691–1695.
- [13] A. Shamir, "How to share a secret," *Commun. ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979. [Online]. Available: <http://doi.acm.org/10.1145/359168.359176>
- [14] S. P. Shariatpanahi, S. A. Motahari, and B. H. Khalaj, "Multi-server coded caching," *arXiv:1503.00265 [cs.IT]*, 2015.
- [15] S. Wang, W. Li, X. Tian, and H. Liu, "Fundamental limits of heterogeneous cache," *arXiv:1504.01123 [cs.IT]*, 2015.
- [16] D. Wessels, *Web Caching*, N. Torkington, Ed. O'Reilly, 2001.
- [17] J. Zhang, X. Lin, C.-C. Wang, and X. Wang, "Coded caching for files with distinct file sizes," in *Proc. IEEE ISIT*, June 2015, pp. 1686–1690.